Cómo crear un live CD de cualquier distro.

Por Haylem Candelario Bauzá. Documento libre.

Crear un live CD es uno de los temas más intrigantes entre los usuarios avanzados de GNU/Linux. Todos queremos tener nuestro sistema personalizado y no tener que repetir el proceso cada vez que reinstalamos una PC y desearíamos poder incluso usar nuestro sistema sin necesidad de usar un disco duro.

Un live CD nos permite arrancar uno de esos sistemas mágicos que lo hacen todo en muy tan poco espacio y hace que nos preguntemos: ¿Cómo se puede hacer todo eso con tan poco?. Mediante este mecanismo seremos capaces de utilizar un sistema operativo insertando un CD/DVD/Blue-Ray/USB/Tarjeta MicroSD, etc. con tan solo insertarlo en el puerto correspondiente de nuestro ordenador previamente configurado para arrancar por estos medios lo cual abre las puertas a miles de posibilidades incluso como servidor congelado o sólo lectura puesto que la mayoría de las veces las imágenes del sistema son de este tipo impidiendo romper el sistema de archivos.

Configurando el BIOS del PC.

Para que nuestro sistema arranque debemos decirle al ordenador que no queremos iniciar por un disco duro de la forma tradicional, si no que queremos que al encenderlo se desvíe hacia una USB u otro medio. Para ello debemos encender el PC y oprimir repetidamente la tecla **DEL** en la mayoría de las computadoras; en otros modelos como las HP se usa la tecla **F10** y en otras se usa **ESC.**

Nos debe aparecer una pantalla estilo años 80; en algunos casos aparece un sistema de ventanas moderno con puntero de mouse. El objetivo aquí, es encontrar las opciones que tienen que ver con el arranque del ordenador que la mayoría de las veces está siempre al final, ahí cambiaremos el orden buscando nuestro dispositivo y subiéndolo en la lista a la primera posición; casi siempre con la tecla **F6**. Luego guardamos la configuración con **F10** y al iniciar iniciará por ese dispositivo en particular.

Creando el liveCD

Para crear nuestro live CD instalaremos un sistema GNU/Linux en una partición o máquina virtual. En este último caso luego se debe montar el disco virtual para empaquetar el sistema de archivos.

Luego de instalar el sistema que deseamos, en este caso usaré Debian cualquier versión, debemos procurar determinar a qué público vamos a dirigir nuestra distro, si es general recomiendo sólo poner programas generales, es decir:

1- Reproductor de música.
2- Un reproductor de video.
3- Juegos buscaminas, solitario, lo básico.
4- Un explorador de archivos.
5- Un compresor de archivos.
6- Una suite de oficina.
7- Editor de video sencillo estilo movie maker como puede ser OpenShot.
8 - Creador/Editor de particiones como gparted.
9- Navegador web.
10- Para compartir datos por la red con protocolo windows como samba.

Luego de instalar nuestros programas, recomiendo configurarlos con una configuración típica que se sepa que funciona para los usuario de manera que estos no necesiten tocar la configuración. Por ejemplo yo siempre pongo una carpeta vacía en /opt la cual uso para configurarla en samba de manera que no haya que configurar luego samba por parte de los usuarios y tengan ya una carpeta lista para compartir con su contraseña. Vale una nota aquí, para activar un usuario de samba se utiliza

smbpasswd -a usuario, y luego debe darle de alta con smbpasswd -e usuario donde usuario es cualquier usuario disponible en el sistema.

Debemos luego configurar nuestro escritorio que debería tener una opción para cambiar el idioma del teclado en inglés o español, un fondo neutro con un tema neutro que no favorezca un tema en particular, podría ser un fondo abstracto por ejemplo o un tema de triángulos con colores llamativos en diseño y que dé la impresión de respeto y seriedad ante el trabajo aunque esto es a gusto del que crea el live CD.

Luego de configurarlo todo, debemos hacer que cada vez que se crea un usuario esa configuración se replique en ellos así como la configuración de los programas de usuario y todas las disposiciones que ya pusimos. Esto se resuelve copiando todos los archivos de nuestro usuario, por ejemplo de la carpeta */home/miusuario/ a la carpeta /etc/skel/ sucede que también debemos tener en cuenta copiar los archivos ocultos, yo uso esto:*

cp -rfv /home/miusuario/ /etc/skel/*
cp -rfv /home/miusuario/. /etc/skel/*

Recomiendo limpiar los residuos de los programas para aligerar nuestro live CD con:

rm -rv */home/miusuario/.cache*
rm -rv /etc/skel/.cache

Si no se hace lo anterior puede que hayan problemas al crear nuevos usuarios.

Luego debemos limpiar los residuos de las conexiones ssh y si usamos aplicaciones de windows limpiarlas también a no ser que querramos que en el live CD esté disponible dicho software, por ejemplo si está haciendo un live Cd con un menú que corre juegos de windows, o sea uno dedicado a jugar.

Debemos limpiar también idiomas innecesarios y borrar los /usr/share/doc/* si desea aligerar aún más la ISO. Ojo, xmaxima requiere que la documentación esté ahí, si usa este programa copie la carpeta maxima-doc de la carpeta doc a /tmp, borre los docs y restaure luego dicha carpeta. Para limpiar más archivos innecesarios recomiendo usar bleachbit

Esta limpieza la podemos llevar a cabo mediante las siguientes órdenes:

rm -rv */home/miusuario/.wine*
rm -rv /home/miusuario/.ssh

Cuando se utilizan aplicaciones windows es común que estas guarden algún tipo de configuración en el escritorio como un acceso directo o bién el icono de cómo aparece en los menús. Estos generalmente se encuentran en la carpeta personal dentro de .local/share/ applications. Recomiendo borrar el contenido de esa carpeta para luego de borrar el directorio .wine no se queden en los menús.

Recomiendo además instalar la mayoría de drivers para dispositivos wifi y todos los paquetes de xserver-xorg-video* menos el nvidia a no ser que no de conflicto al estar instalado junto a los demás. Si instala los drivers de nvidia junto con los demás sabiendo que generan conflicto podría quedarse sin aceleración gráfica de momento haciendo que todo el sistema funcione más lento e impidiendo que algunas aplicaciones gráficas no se ejecuten como puede ser blender. Así mismo instale todos los drivers de impresoras disponibles en el repo. Debería distribuir además todos los codecs de videos y audio libres. Para encontrar todas estas cosas puede buscar en su administrador de programas con términos como codecs, drivers, firmware, etc y leyendo la descripción del paquete para ver si conviene instalarlo o no.

Si su liveCD es para 32 bits, recomiendo luego de instalarlo todo y configurarlo todo instalar como último paso un kernel a 64 bits aparte del de 32 bits. De manera que pueda aprovechar toda la ram en sistemas más modernos, luego se creará un menú en el live CD para usar un kernel u otro al arrancar. En sistemas de 32 bits debería hacer que el kernel por defecto sea el de 32 bits para evitar que el sistema no bootee al iniciarlo en una PC antigua, déje el kernel de 64 bits como segunda opción si la prioridad es 32 bits.

Bueno, ya tenemos todo listo para empaquetar. Antes de hacer esto debemos contemplar algunos detalles, en el archivo **/etc/fstab** se guarda la información de aranque de los discos y en qué órden se montan al iniciar el sistema, debemos eliminar todo el contenido del archivo dejando sólo el apartado de montar el cd sr0.

Esto lo hacemos desde otro sistema GNU/Linux en modo live CD o bién desde otra partición con GNU/Linux. Debemos además instalar como último paso el paquete live-boot que contiene scripts que hacen que el sistema se adapte en cada PC que bootea cargando automáticamente lo que necesita en cada caso particular.

apt install live-boot en Debian y derivados.

Esto actualizará el initramfs agregando dichos scripts incluso firmwares.

Estando en otro sistema GNU debemos tener instalado en él la herramienta mksquashfs que se encuentra en el paquete squashfs-tools, instálelo con:

apt install squashfs-tools por ejemplo o bien puede copiarlo de la carpeta:

para-mi-live/usr/bin/mksquashfs

a ***/usr/local/bin/*** en el live CD actual o sistema actual, esto para no tener que instalarlo.

Ya estamos listos, vamos a crear nuestra carpeta de trabajo:

mkdir miDistro

Entramos en ella:

cd miDistro

Creamos una carpeta para montar nuestro sistema a crear:

mkdir sistema

Ahora montamos el sistema que hemos configurado en ella:

mount /dev/sda1 ./sistema

Si listamos la carpeta **sistema** con el comando ***ls*** podemos ver una estructura de sistema de archivo UNIX con bin, etc, usr, etc.

Ahora debemos eliminar cualquier cosa dentro de ***./miDistr/media/*,*** esto es debido a que el sistema operativo cuando monta una usb, disco esterno, partición, etc crea una carpeta con la misma etiqueta de dispositivo por ejemplo **memoriaRoja** y establece los permisos de el usuario actual a dicha carpeta. Si omitimos este paso es probable que no tengamos permisos para montar particiones en nuestro live CD ni luego de instalar hasta que borremos ese contenido, mostrando un mensaje de permiso denegado.

Ahora sí estamos listos!!!!

teclee en la terminal actual:

mksquashfs ./sistema filesystem.squashfs -comp xz

Esto creará una imágen de disco en formato squashfs comprimida con xz, por sí misma no

tiene utilidad alguna pero se podría usar para restaurarla en otro disco montando ese squashfs con: mount -o loop ./filesystem.squashfs ./puntoDeMontaje y copiando todo su contenido a una partición vacía. Básicamente este sería el principio de un instalador al disco duro.

Ya tenemos nuestro sistema listo para ser booteado pero debemos crear una especie de estructura para que el sistema pueda bootearlo. Para eso vamos a crear otra carpeta en otra localización con la siguiente estructura:

Carpeta boot
Carpeta live
    Archivo kernel
    Archivo initrd
    filesystem.squashfs

Para obtener el kernel debemos usar exáctamente el del sistema que hemos configurado.

Creando todo.

mkdir micd
mkdir live
mkdir boot

cp -v ./sistema/boot/vmlinuz* ./micd/live/kernel

cp -v ./sistema/boot/initramfs* ./micd/live/initramfs

Creando un menú de booteo.

Ahora llega el momento de hacerle un menú de arranque a nuestra distro, lo podemos hacer con grub pero el estándar para esto es isolinux.

Debemos instalar isolinux con:

apt install isolinux

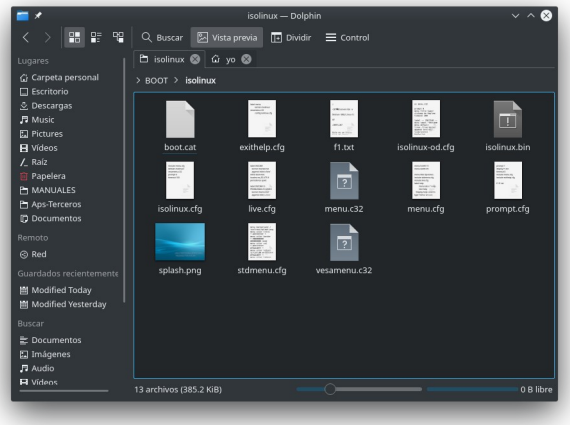
Isolinux es una versión de syslinux para formato ISO o CD, también existe pxelinux para red, extlinux para ext3/4, syslinux para fat32.

La forma de trabajar de syslinux es que cuando el iso bootea o el CD, se busca el archivo isolinux.cfg en caso de isolinux o syslinux.cfg en caso de syslinux, este último para tener el iso en una usb. Este archivo es de texto plano y contiene la entradas de menú, imágen de fondo , etc de nuestro live CD. Cada entrada de menú indica un kernel con unos parámetros a cargar, así de simple. También en boot debe haber isolinux.c32 e isolinux.bin que son los mbr que llamará a isolinux al bootear el disco, estos archivos son binarios y puede obtenerlos de otro livecd montando la iso o bién puede copiarlos de /usr/share/isolinux a su carpeta boot del live CD y sólo tendría que editar el isolinux.cfg con algo como esto:

```
label INICIAR
  kernel /live/kernel
  append initrd=/live/initrd boot=live locales=es_ES.UTF-8 quiet

label INICIAR-SI-PROBLEMAS-EN-VIDEO
  kernel /live/kernel
  append initrd=/live/initrd boot=live locales=es_ES.UTF-8  quiet

label INICIAR-MODO-CONSOLA-RECOVER
  kernel /live/kernel
append initrd=/live/initrd boot=live locales=es_ES.UTF-8 quiet
```



Para usar la persistencia puede agregar la opción ***persistence***

También está disponible el archivo vesamenu.c32 que al cargarse permite poner una imágen de fondo en el live CD. Estos archivos .c32 se pueden considerar como micro-kernels que se cargan y hacen una función.

Para una configuración con vesamenu en el isolinux.cfg:

include menu.cfg
default /isolinux/vesamenu.c32
prompt 0
timeout 100

El timeout es el tiempo de espera si no se oprime nada. En este caso se bootea por la opción marcada como default. Si usamos isolinux en este caso con vesamenú de esta forma debemos definir el menú en menu.cfg.

menu hshift 15

menu width 49

menu title Opciones.
include stdmenu.cfg
include live.cfg
label help
menu label ^Help
text help
Display help screens; type 'menu' at boot prompt to return to this menu
endtext
config prompt.cfg

Vemos que se hace referencia a live.cfg que sería nuestro menú. EL ^ Es para marcar la tecla aceleradora, y los text pueden ser archivos txt por ejemplo F1.txt que se llaman al oprimir las teclas de función ***F1-F12*** y su finalidad es mostrar la ayuda. O sea que en esos archivos guarde una descripción en texto plano o una ayuda. Puede usar colores.

Y el live.cfg contendría la descripción de menú vista al inicio.

Quiero mi iso!!! dónde está mi ISO!!!.

Listo amigos!!! solo nos queda crear nuestro iso para quemarlo en nuestro CD dvd. Para ello usamos el comando genisoimage, no se asusten:

apt install genisoimage (para instalar el comando en Debian)

Y empaquetamos:

genisoimage -v -r -V "La_Distro_Fast" -cache-inodes -J -l -b isolinux/isolinux.bin -c isolinux/boot.cat -no-emul-boot -boot-load-size 4 -boot-info-table -o remaster.iso micd/

Observen por favor:
-V Indica la etiqueta del CD, con -b cargamos el sector que se quemará en el disco, el mbr del disco, si no no bootearía. Con -c el archivo catálogo de booteo tiene que ver con la estructura del booteo del disco. Las opciones -boot* noemul* hacen que no se use la emulación de disco duro, en algunas Pcs antiguas puede ser que requiera esto pero yo nunca lo he necesitado, nunca me ha dado problemas esto, -o especifica el nombre del iso final y ***micd*** recuerden que es la carpeta donde se está haciendo todo, o sea la que contiene el directorio boot y live.

Ya tiene su iso listo para quemar o probar en una PC virtual. Si en una PC virtual ocurriera algún problema de no encontrar el FS es porque no está viendo el filesystem.squashfs, el kernel contiene el módulo que entiende este formato al bootear por lo que lo primero que busca es un archivo con nombre filesystem.squashfs y si lo encuentra lo carga, usualmente dentro de una

carpeta live. Se puede resolver este error que da en sistemas antiguos pasándole al kernel la opción root=/dev/sda1, donde sda1 es una partición del disco , esto para que el live pueda volcar datos temporales. Si la partición contiene windows no se dañará no se preocupe. Esto sólo lo he visto necesario hacer en Debian 8 con qemu pero al parecer era un error en qemu y el tener instalado live-config junto con live-boot.

Todo bien!!! pero… lo quiero ahora en mi usb sin usar programas.

Hay!!! estos usuarios son como los roedores, intranquilos. Está bien, sólo debemos convertir nuestra querida iso a modo híbrido para que se comporte también como una iso de disco duro, esto no afectará que se pueda quemar en un CD. Al hacer esto se puede pasar el iso a una usb directamente con el comando dd.

Para esto usaremos el comando isohybrid del paquete syslinux-utils. Para instalarlo en Debian:

apt install syslinux-utils

Le pasamos el comando a nuestro iso.

isohybrid ./remaster.iso

Y listo!!!, ahora puede pasar el iso a cualquier usb con:

dd if=./remaster.iso of=/dev/sdb1 bs=1M conv=fdatasync

If indica archivo de entrada, of es la salida, bs es el tamaño de bloque, recomiendo 1M, y conv se asegura que se pase sincronizado de otro modo no me ha funcionado.

Esto de isohybrid es para imágenes de GNU/Linux, las de windows necesitan otro método por el formato.

Para montar un disco qcow/qcow2 el cuál contiene la estructura de su live cd debe cargar el modulo nbd

modprobe nbd

y luego montarlo con:

qemu-nbd -c /dev/nbd0 ./discoVirtual.disk

Para desmontarlo:

qemu-nbd -d /dev/nbd0

Para crear un instalador debe lograr que el mismo monte la partición de un disco duro en una carpeta y luego copie el contenido del filesystem.squashfs el cual debe montar en otra carpeta. Debe copiar el contenido con rsync -av punto de montajeSquash/ y cópielo a puntoDeMontajeDiscoDuro/. Generalmente los demás archivos del USB o disco live y sus carpetas boot y live no se ven en el árbol principal de directorios pero puede ver e incluso agregar documentos a su memoria en la estructura de archivos /lib/live/…*

Luego debe decirle al instalador que escriba el comando para instalar grub en el .bashrc de su sistema pasado al disco duro, luego debe hacer chroot a ese punto de montaje para que se ejecute dicho comando, actualizar el grub desde el mismo .bashrc con update-grub que es otra órden, eliminar las líneas de instalación/actualización del grub del .bashrc ,salir del chroot, desmontar el disco duro y reiniciar.

De esta manera usted puede crear su propia distro para conservar programas que podrían desaparecer incluso, y que por nostalgia quisiera usar luego de 20 años o más. Saludos.

Anexo. Instalador para liveCD Debian.

```
#!/bin/sh

# Instalador de Sistem Debian modo texto
# Escrito por Haylem Cadelario Bauza

SEL=""
FMT="ext4"
PAR="sda"
SWAP="sda2"
NUM="1"

clear
echo && echo && echo

# Seleccion del formato de instalacion

echo "     Seleccione el formato del disco destino:"
echo && echo && echo
echo "      1)- ext3"
echo "      2)- ext4"
echo "      3)- ReiserFS"
echo "      4)- XFS"
echo

echo -n "Introduzca su seleccion--> "
read SEL

if [ -z $SEL ];then
  echo "Usando seleccion por defecto $FMT"
  SEL=0
fi

if [ $SEL -eq 1 ];then
  FMT="ext3"
fi

if [ $SEL -eq 2 ];then
  FMT="ext4"
fi

if [ $SEL -eq 3 ];then
  FMT="reiserfs"
fi

if [ $SEL -eq 4 ];then
  FMT="xfs"
fi

echo && echo

echo -n "Dispositivo donde instalar el sistema (Ej sda sin el numero): "
read PAR

if [ -z $PAR ];then
  PAR="sda"
fi


echo -n "Numero de particion[1-...] (Ej 1): "
read NUM

if [ -z $NUM ];then
  NUM="1"
fi

PAR="/dev/$PAR"

echo && echo

echo -n "Particion swap (Ej sda2): "
read SWAP

if [ -z $SWAP ];then
  SWAP="sda2"
fi

SWAP="/dev/$SWAP"

echo && echo

echo "|--OK->"
echo
echo "   El sistema se va a instalar usando los siguientes datos:"
echo
echo "   La particion $PAR$NUM se va a formatear"
echo "   usando el formato $FMT. El sistema va a usar"
echo "   la particion $SWAP como intercambio."
echo && echo


echo -n "Presione una tecla para continuar o ctrl+c para cancelar..."
&& read tmp

clear


# Crear estructura-----------------


umount /target
umount /mnt2
umount $PAR$NUM

mkdir /target
mkdir /mnt2

# Formatear y preparar----------

echo
echo "Formateando la particion $PAR$NUM con formato $FMT..."
mkfs.$FMT $PAR$NUM
tune2fs -r 5000  $PAR$NUM

echo "Montando $PAR$NUM en /target"
mount $PAR$NUM  /target

mount -o loop -t squashfs /live/image/live/filesystem.squashfs /mnt2
mount -o loop -t squashfs
/lib/live/mount/medium/live/filesystem.squashfs /mnt2




echo
echo "Copiando el sistema, sea paciente..."


#cp -dpRv /mnt2/* /target/
#./copia -av /mnt2/* /target/ --exclude-from=./instalador.list
rsync  -av  /lib/live/mount/rootfs/filesystem.squashfs/*  /target/  --exclude-
from=./instalador.list

echo && echo
echo "Configurando el sistema..."

cp /target/usr/sbin/update-initramfs.distrib /target/usr/sbin/update-
initramfs
cp /target/usr/sbin/update-initramfs.debian /target/usr/sbin/update-
initramfs

# Borrar archivos innecesarios...
rm -rv /target/media/*
rm -rv /target/mnt/*
rm -rv /target/sys/*
```

```
rm -rv /target/proc/*
rm -rv /target/tmp/*
rm -rv /target/live
rm -rv /target/var/lib/live
rm -rv /target/boot/grub
rm -rv /target/etc/udev/rules.d/70-persistent-cd.rules
rm -rv /target/etc/udev/rules.d/70-persistent-net.rules


rm -v /swapfile
rm -v /target/swapfile
rm -v /etc/fstab
rm -v /etc/mtab

touch /etc/fstab
touch /etc/mtab

# Preparar para instalar grub

mount  --bind /dev/ /target/dev/
mount  --bind /sys/ /target/sys/
mount  --bind /proc/ /target/proc/
echo "proc/proc	proc	defaults	0	0"	>
/target/etc/fstab
echo "$PAR	/	$FMT	defaults,noatime	0
	1" >> /target/etc/fstab
echo "$SWAP	none	swap	sw	0	0" >>
/target/etc/fstab

grep -v user /etc/sudoers > /target/etc/sudoers

cp /target/root/.bashrc /target/root/.bashrc.orig

echo "grub-install --force --recheck --no-floppy --root-directory=/
$PAR">> /target/root/.bashrc
echo "update-grub" >> /target/root/.bashrc


 echo "adduser usuario cdrom "> /target/root/.bashrc
 echo " adduser usuario floppy"> /target/root/.bashrc
 echo " adduser usuario audio"> /target/root/.bashrc
 echo " adduser usuario dip"> /target/root/.bashrc
 echo " adduser usuario video"> /target/root/.bashrc
 echo " adduser usuario plugdev"> /target/root/.bashrc
 echo " adduser usuario users"> /target/root/.bashrc
 echo " adduser usuario powerdev"> /target/root/.bashrc
 echo " adduser usuario scanner"> /target/root/.bashrc
 echo " adduser usuario netdev"> /target/root/.bashrc
 echo " adduser usuario dialout"> /target/root/.bashrc
 echo " adduser usuario fuse"> /target/root/.bashrc
 echo " adduser usuario haldaemon"> /target/root/.bashrc
 echo "dpkg -r instalador"> /target/root/.bashrc

echo "mv /root/.bashrc.orig /root/.bashrc" >> /target/root/.bashrc
echo "exit" >> /target/root/.bashrc



# Esto hace que se ejecute el instalador de grub al ejecutar
# Automaticamente .bashrc de la carpeta root

chroot /target/ # Se ejecuta lo que hay dentro de .bashrc

# Desmonta particiones a la fuerza -l

umount -l /target/dev
umount -l /target
umount -l $PAR$NUM

reboot # Reinicia al terminar
```

Exclusiones instalador.list

```
/mnt2/dev/*
/mnt2/cdrom/*
/mnt2/media/*
/mnt2/target
/mnt2/swapfile
/mnt2/mnt/*
/mnt2/sys/*
/mnt2/proc/*
/mnt2/tmp/*
/mnt2/live
/mnt2/var/lib/live
/mnt2/boot/grub
/mnt2/etc/udev/rules.d/70-persistent-cd.rules
/mnt2/etc/udev/rules.d/70-persistent-net.rules
```

Nota: Si no excluye live se copiará el sistema recursivamente de modo indefinido.